**INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA DE MINAS GERAIS – CAMPUS SÃO JOÃO EVANGELISTA**

**BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

GABRIEL KÁICON BATISTA HILÁRIO

**TRABALHO PRÁTICO I**

**SÃO JOÃO EVANGELISTA**

**SETEMBRO - 2022**

GABRIEL KÁICON BATISTA HILÁRIO

SISTEMA DE CADASTRO DE FUNCIONÁRIOS E DE PROJETOS DENTRO DE FUNCIONÁRIOS

**SÃO JOÃO EVANGELISTA**

**SETEMBRO - 2022**

SUMÁRIO

[1. INTRODUÇÃO 9](#_Toc115137691)

[1.1. Objetivo Geral 9](#_Toc115137692)

[1.2. Objetivos Específicos 9](#_Toc115137693)

[1.3. Justificativa 9](#_Toc115137694)

[2. DESENVOLVIMENTO 11](#_Toc115137695)

[2.1. Conceitos: Lista Encadeada, Lista sequencial, TADs e arquivos 11](#_Toc115137696)

[2.2. Implementação 13](#_Toc115137697)

[3. CONCLUSÃO 22](#_Toc115137698)

[4. REFERÊNCIAS 23](#_Toc115137702)

[5. APÊNDICES 24](#_Toc115137703)

[5.1. APÊNDICE A Funções para manipulação de lista Encadeada 24](#_Toc115137704)

[5.2. APÊNDICE B – Funções para a manipulação de Lista Sequencial 25](#_Toc115137705)

[5.3. APÊNDICE C – Sistema.hpp 27](#_Toc115137708)

[5.4. APÊNDICE D – Criação de funcionário 27](#_Toc115137710)

[5.5. APÊNDICE E – Pesquisa ID repetido 27](#_Toc115137712)

[5.6. APÊNDICE F – Função de criação de projeto 28](#_Toc115137714)

[5.7. APÊNDICE G – Cálculo de Horas semanais trabalhadas. 28](#_Toc115137716)

[5.8. APÊNDICE H – Cálculo de Salário Bruto 28](#_Toc115137718)

# INTRODUÇÃO

Este trabalho prático foi documentado para que seja avaliado em conjunto com os códigos na linguagem C/C++, exigido pelo docente Eduardo Augusto da Costa Trindade, dentro da disciplina de Algoritmos e Estruturas de Dados I, ministrada pelo mesmo. Porém a documentação tem cunho expositivo, onde é descrito as funcionalidades do programa, com testes, e desenvolvimento de novas linhas de raciocínio lógico para realização do trabalho prático.

## Objetivo Geral

Este trabalho tem como objetivo geral apresentar na prática os conhecimentos adquiridos nas aulas de Algoritmos e Estruturas de Dados I, a respeito de Listas com Arranjo, Listas Encadeadas, Tipos Abstratos de Dados (TADs), Manipulação de Arquivos e Ponteiros, utilizando a linguagem de C/C++ para escrita dos códigos.

## Objetivos Específicos

Esse trabalho tem como objetivos específicos:

* Apresentar conhecimentos em lista encadeada;
* Apresentar conhecimentos em lista com arranjo;
* Apresentar conhecimentos em manipulação de arquivos;
* Aplicar conhecimentos adquiridos em um minissistema, em uma visão empresarial.

## Justificativa

Quando se aprende sobre Estruturas de Dados, vemos o conteúdo a respeito de Ponteiros, e o conteúdo de Manipulação de Arquivos, vemos o *ifstream*(usado para a leitura de um arquivo) e o *ofstream*(usado para escrever em um arquivo), ambos são objetos básicos da biblioteca *fstream*, que tem finalidade de manipulação de arquivos. O código inteiro foi escrito na linguagem C/C++, com maior foco na linguagem C++.

Posteriormente vemos o conteúdo Estruturas de Dados mesmo, a começar pela estrutura de Listas, essas listas possuem itens, e o que varia dessas listas, são os tipos de itens que serão inseridos. Inicialmente nos é apresentado dois tipos de lista.

* A lista sequencial que é com alocação estática, ou seja, possui um limite pré-definido, podendo ter um número limitado de itens.
* A lista encadeada com alocação dinâmica, ou seja, não possui um limite pré-definido podendo ter um número infinito de itens.

A princípio são aplicados de forma individual, porém no trabalho, devemos cadastrar funcionários em uma lista encadeada, e devemos atribuir um ou mais projetos já cadastrados em uma lista sequencial a esse funcionário que está na lista encadeada.

Os 4 conteúdos foram aplicados para realização do trabalho, criando um minissistema que nos permite cadastrar e excluir funcionários, atribuir e retirar projetos ao funcionário, e calcular e imprimir o contracheque.

# DESENVOLVIMENTO

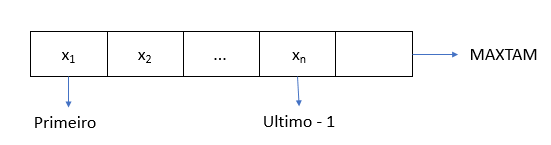
Nesta seção do documento é apresentado o desenvolvimento do trabalho.

## Conceitos: Lista Encadeada, Lista sequencial, TADs e arquivos

Aqui falo de forma bem superficial o que contém no meu programa.

Sobre as TADs, Tipos Abstratos de Dados, em C++ às declaramos como structs, ou estruturas de Dados acompanhados com funções para manipular a mesma. Structs são representações de qualquer coisa no mundo real, sendo ela lógica ou física, como por exemplo uma pessoa, que é algo físico, ou um filme digital, cada um com suas características específicas. Uma pessoa tem sexo, idade, nome, altura, dentre outras, e um filme tem o elenco, personagens, duração, categoria, ano de lançamento, dentre outros, e tudo isso poderia estar dentro de uma struct definida para cada um deles. Resumindo uma Struct é uma espécie de modelo para cadastrar diferentes itens, que tem características comuns, em uma lista, dentro de um software escrito em C/C++. Acompanhado das structs temos as funções para manipulação dos dados dessa lista, e desses itens. Juntos eles compõem o que chamamos de TAD.

Figura 1 – Lista Sequencial

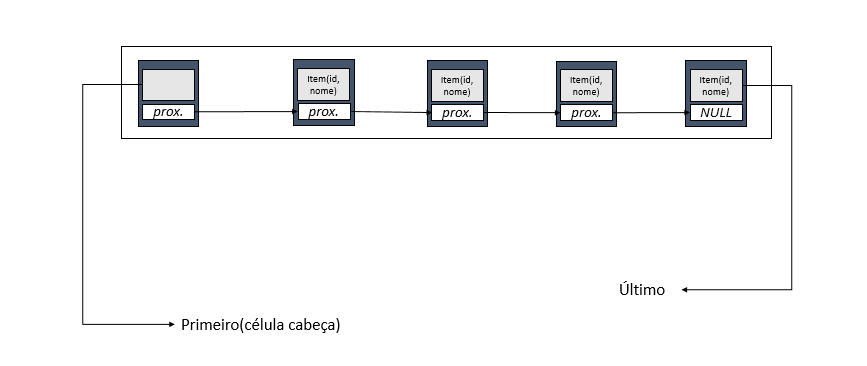


Na figura 1, vemos um desenho esquemático de como seria uma lista sequencial, e o limite dela, é o MAXTAM, uma variável didática muito comum de ser vista. Ela serve para delimitar o tamanho da lista, o “Ultimo” seria o MAXTAM – 1.

Ela é bem semelhante a um vetor, porém a tipagem de dados a ser inserida podem ser as structs que são usadas para representar coisas do mundo real, podendo armazenar mais de um tipo de variável dentro de um único “quadradinho”.

Os elementos são inseridos dentro de um espaço do vetor, ao se inserir novos elementos, eles são alocados no Ultimo, e assim sucessivamente, até que a Lista alcance o tamanho máximo, que seria o valor definido para o MAXTAM.

Figura 2 – Lista Encadeada

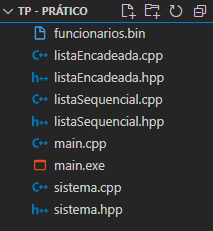


A Figura 2, apresenta um desenho esquemático de como seria uma lista encadeada, e como podemos ver, não há nada delimitando-a. A célula cabeça, que seria o primeiro, não possui nada, apenas um apontador prox, que aponta para o apontador do prox do próximo elemento da lista, e assim sucessivamente, até chegar no elemento que tem apontador NULL, que viria a ser o último elemento, onde novas inserções podem ser feitas. Para realizar uma inserção, dentro desse esquema, se adiciona ao próximo quadradinho depois do último, uma célula e um apontador prox, e dentro dessa célula se coloca o item especificado pela TAD, e logo em seguida se atualiza o apontador do antigo último que era NULL para prox que apontará para o novo último, e nesse novo item adicionado no fim, recebe o apontador prox como NULL. Há a possibilidade de inserir no inicio e no fim da lista, porém dei um foco maior na inserção no fim, pois no trabalho só irei inserir no fim.

A parte de manipulação de arquivos é mais abstrata, não é convencional para ilustrar ela, mas podemos imaginar fazendo uma analogia. Imagine-se com um livro na mão e um caderno, o caderno é seu arquivo, e o livro é seu programa, quando você quer guardar algo importante do livro, você lê o que está nele e escreve no caderno, isso seria semelhante às funções do ofstream, que pega os dados digitados no programa e escreve no arquivo, independente da extensão deste. Imagine novamente, com um caderno e uma folha apenas, o caderno é o arquivo, e a folha é o que você leu dele, tendo algo escrito no caderno, você realiza a leitura e exibe isso para si mesmo, seria semelhante às funções do ifstream, que pega os dados do arquivo e exibe na tela.

## Implementação

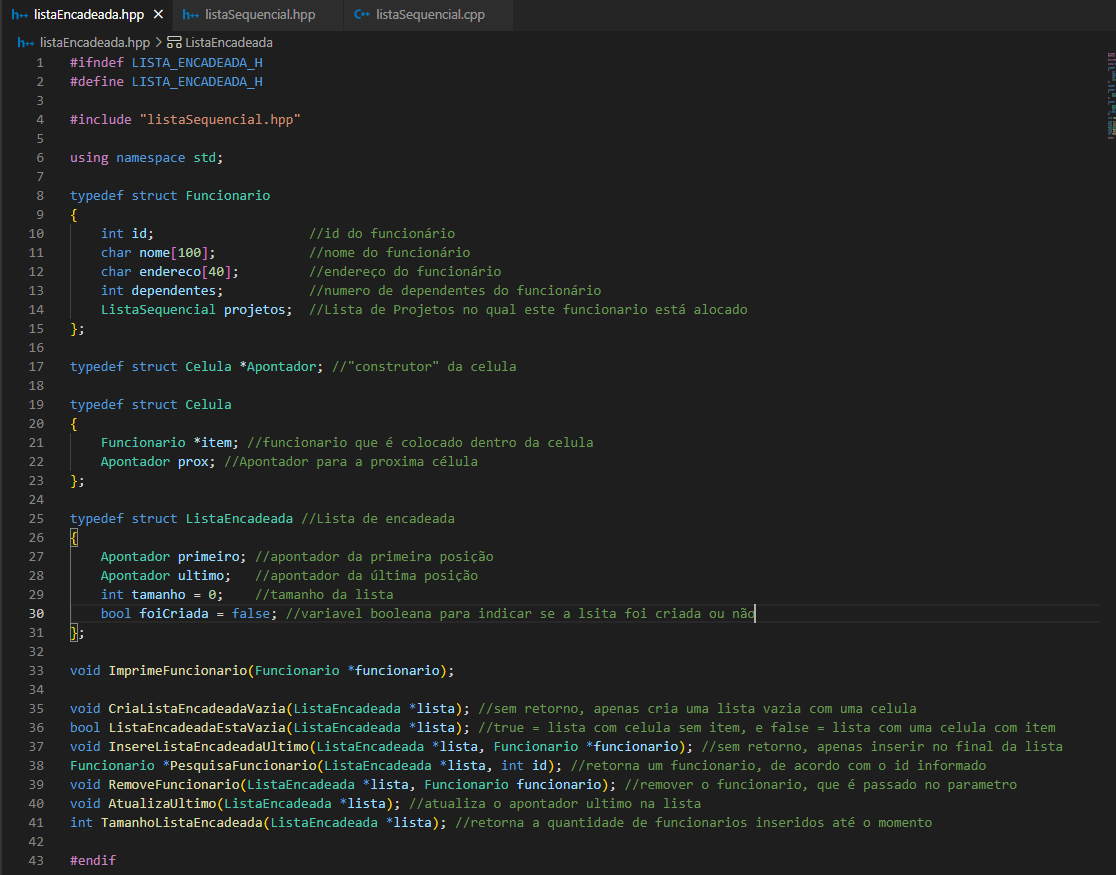
Figura 3



O trabalho foi dividido em 9 arquivos, incluindo a main, e o arquivo bin, como apresentado na figura 3. Com exceção da main, todos os arquivos \*.cpp tem um arquivo cabeça \*.hpp.

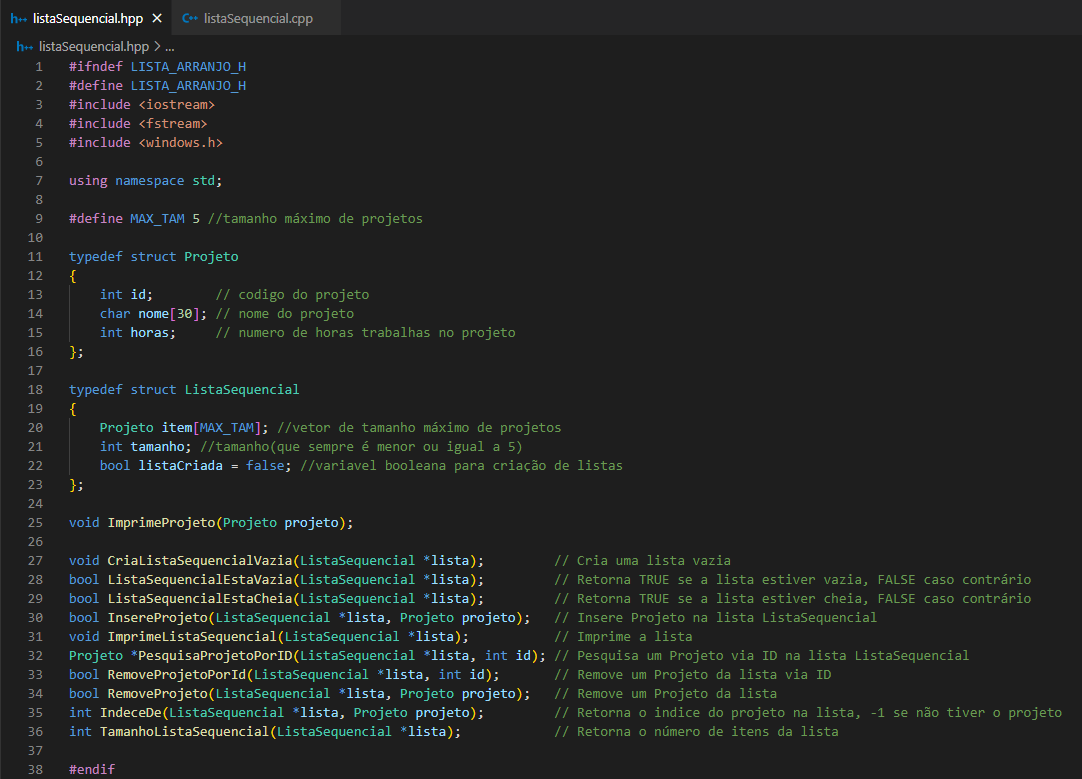
A listaencadeada.cpp, possui as funções básicas para manipulação de TADs(apêndice A), e a listaEncadeada.hpp apresentada na figura 4, possui as TADs.

Figura 4



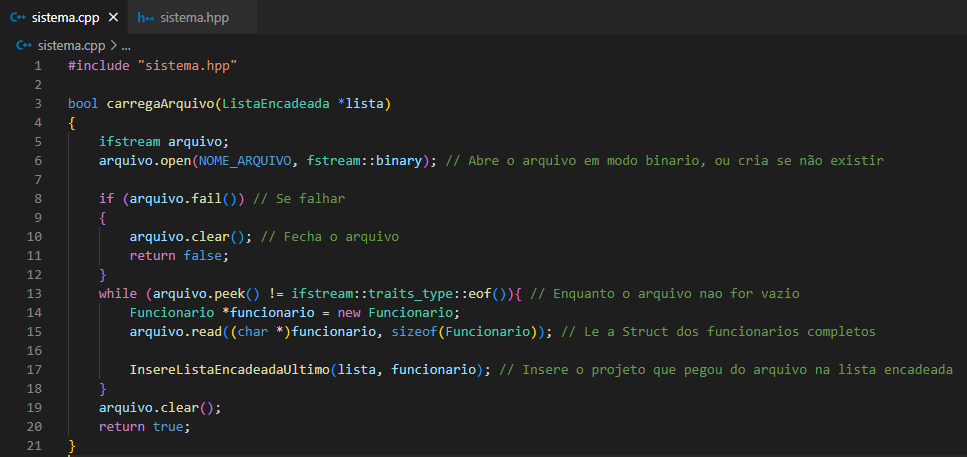
A listaSequencial.cpp, possui funções básicas para manipulação das TADs(Apêndice B), e a listaSequencial.hpp apresentada na figura 5, com as TADs

Figura 5



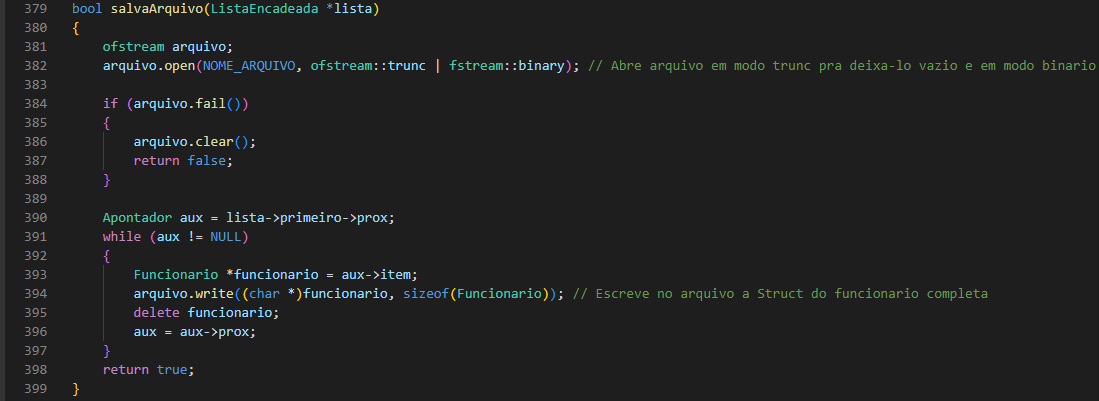
O sistema.cpp, conta com a maioria das funções que são utilizadas na dentro dele e todas da main.cpp, onde é feita toda a manipulação do arquivo para inclusão do funcionário nele, a inclusão e exclusão de projetos ao funcionário, a inclusão e a exclusão do mesmo. E o arquivo sistema.hpp (Apêndice C), com as inclusões gerais de bibliotecas e de outros arquivos \*.hpp, e o cabeçalho das funções para melhor entendimento. Abaixo terão algumas funções de extrema importância, e a devida explicação caso necessário.

Figura 6



Na figura 6, vemos o processo de carregamento do arquivo \*.bin, fazendo validações para evitar erros, e na linha 17, os dados que estavam no arquivo, são inseridos 1 por 1 dentro do laço na lista Encadeada.

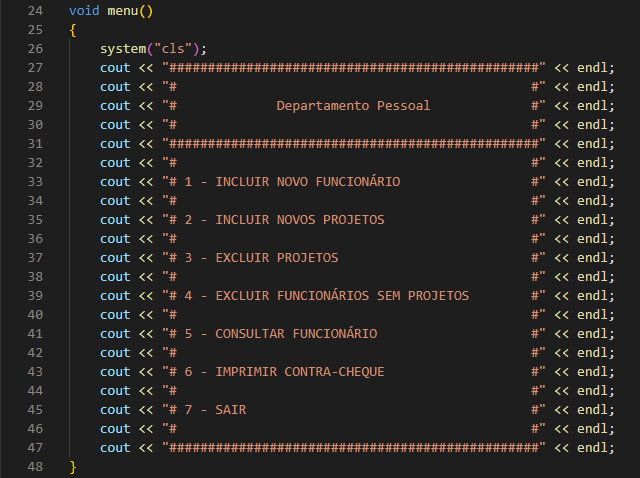
Figura 7



Na figura 7, vemos o processo de salvamento do arquivo, onde o arquivo é aberto vazio, para que possa ser gravado as novas alterações, fazendo os devidos tratamentos para evitar falhas. Vemos o processo de escrita no arquivo na linha 394, e na linha seguinte, de número 395, a memória vai sendo liberando dentro do laço de repetição, a cada escrita.

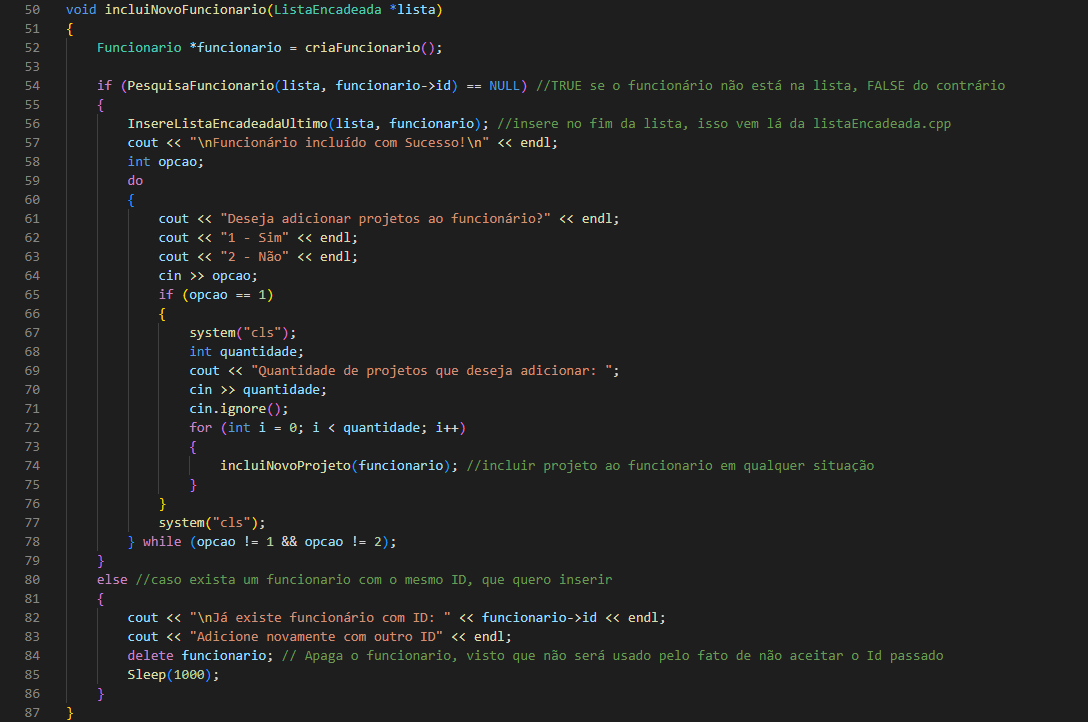
Tendo finalizado a parte que falava de arquivos vamos começar com os cadastros, onde é necessário abstrair muita coisa e dar foco principal nas principais funções.

Figura 8



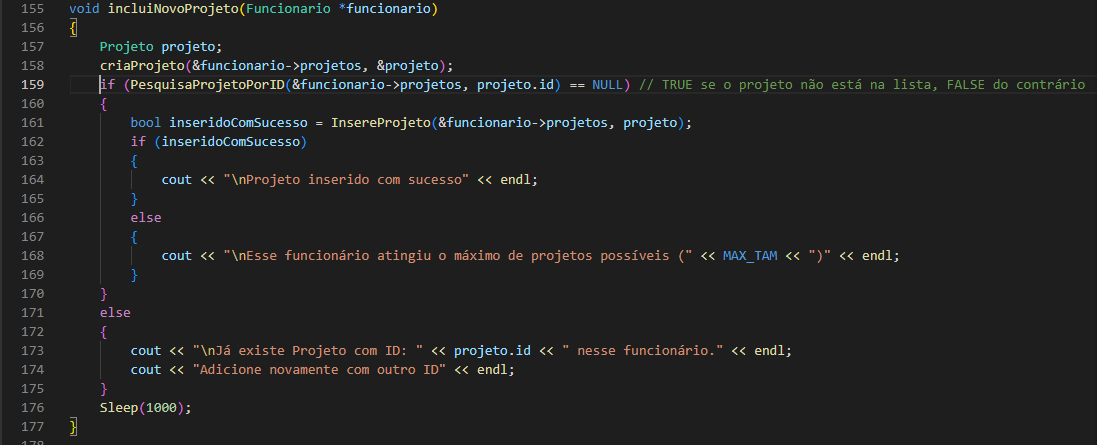
Na figura 8, vemos o menu, com as principais funcionalidades do sistema, a começar pela inclusão do funcionário, podendo incluir ou não projetos nele.

Figura 9



Na figura 9, vemos o processo de inclusão do funcionário. Na linha 52, é chamada uma função (Apêndice D), que “abre” um “menu” para inserir o funcionário, após isso é verificado se o Id inserido para ele já não existe (Apêndice E), se existir ele reinicia o menu, se não ele insere o funcionário, e pergunta se quer que inclua projetos a ele, na linha 74 é chamada uma função (Figura 10), que “abre” um” menu” para inserir projetos a esse funcionário, com a função criaProjeto (Apêndice F) na linha 159. Pesquisa o id do projeto, se der false, significa que pode inserir, e é feita a inserção com a função de insereProjeto (Apêndice B). Essa questão exige muita interpretação, pois inserimos uma lista sequencial dentro de uma parte da célula do funcionário.

Figura 10



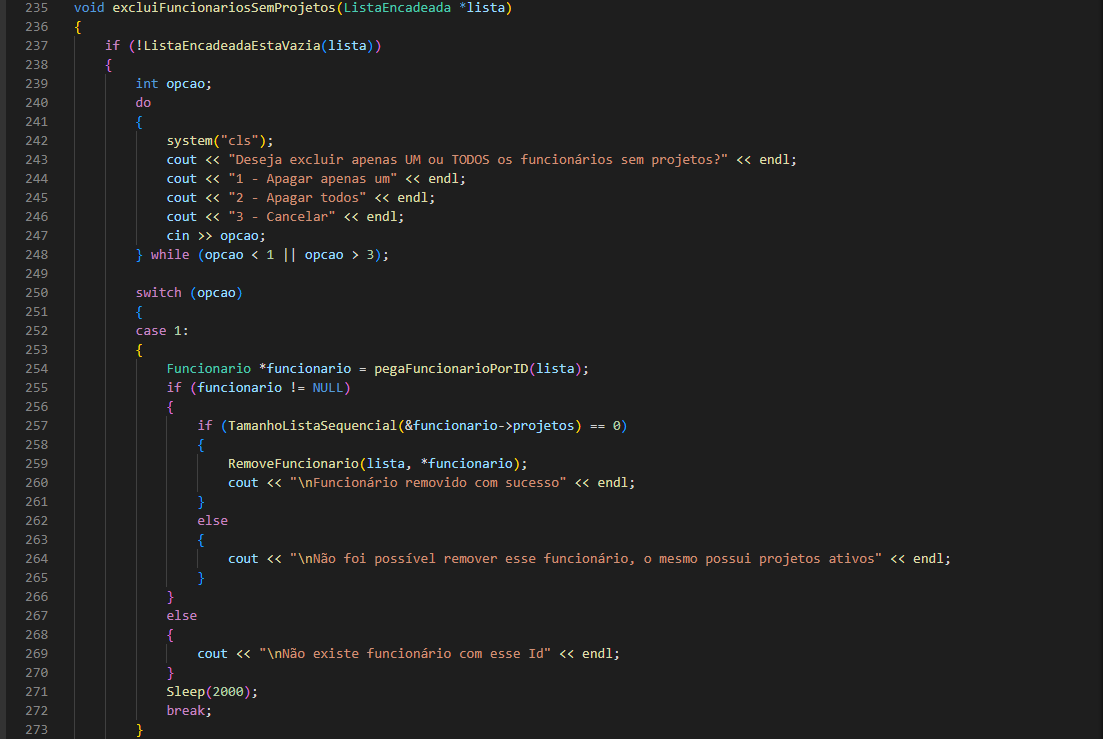
Na figura 11, vemos a função para exclusão de projetos de um funcionário. Onde o funcionário é procurado, para que se possa remover um projeto, também encontrado por meio de Id, e logo em seguida é removido com uma função removeProjeto (Apêndice B).

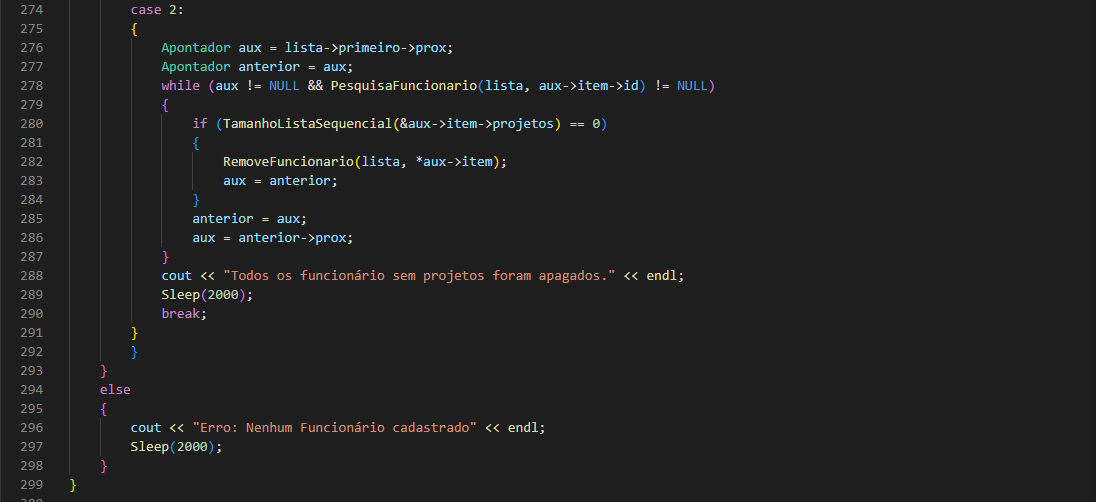
Figura 11



Na figura 12, podemos dar continuidade, e ver a quarta função, que é a função que exclui funcionários sem projetos, podendo excluir 1 ou todos os funcionários sem projetos. O critério para excluir os funcionários sem projetos é simples, basta eles terem a sua lista sequencial com o tamanho = 0, e utilizar a função para excluir o funcionário.

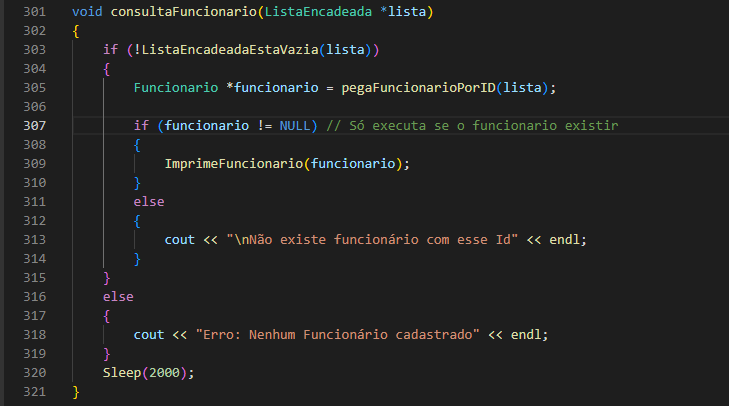
Figura 12





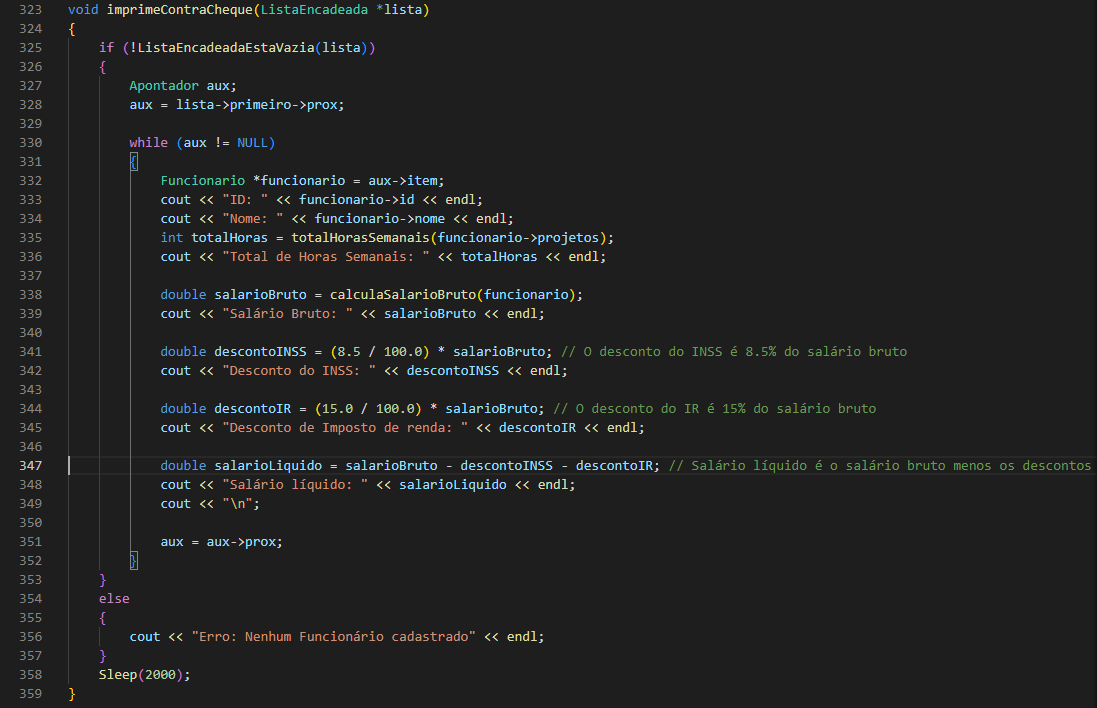
A próxima função apresentada na figura 13, é uma função simples de consulta, que usa da função de procurar o funcionário pelo id(Apêndice E), e imprime ele.

Figura 13



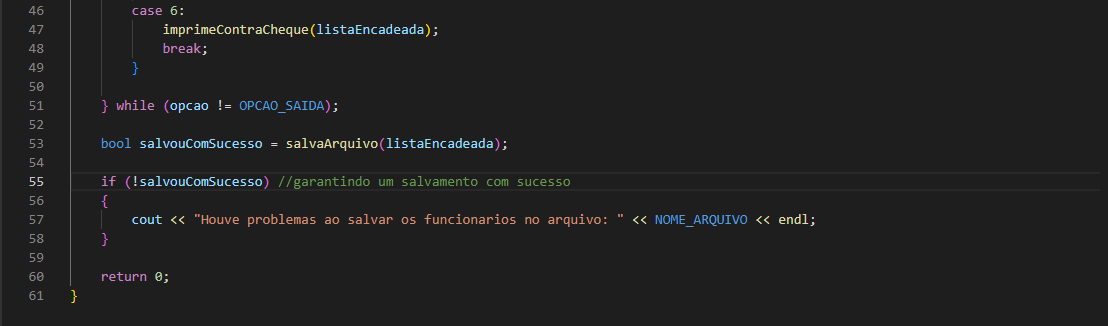
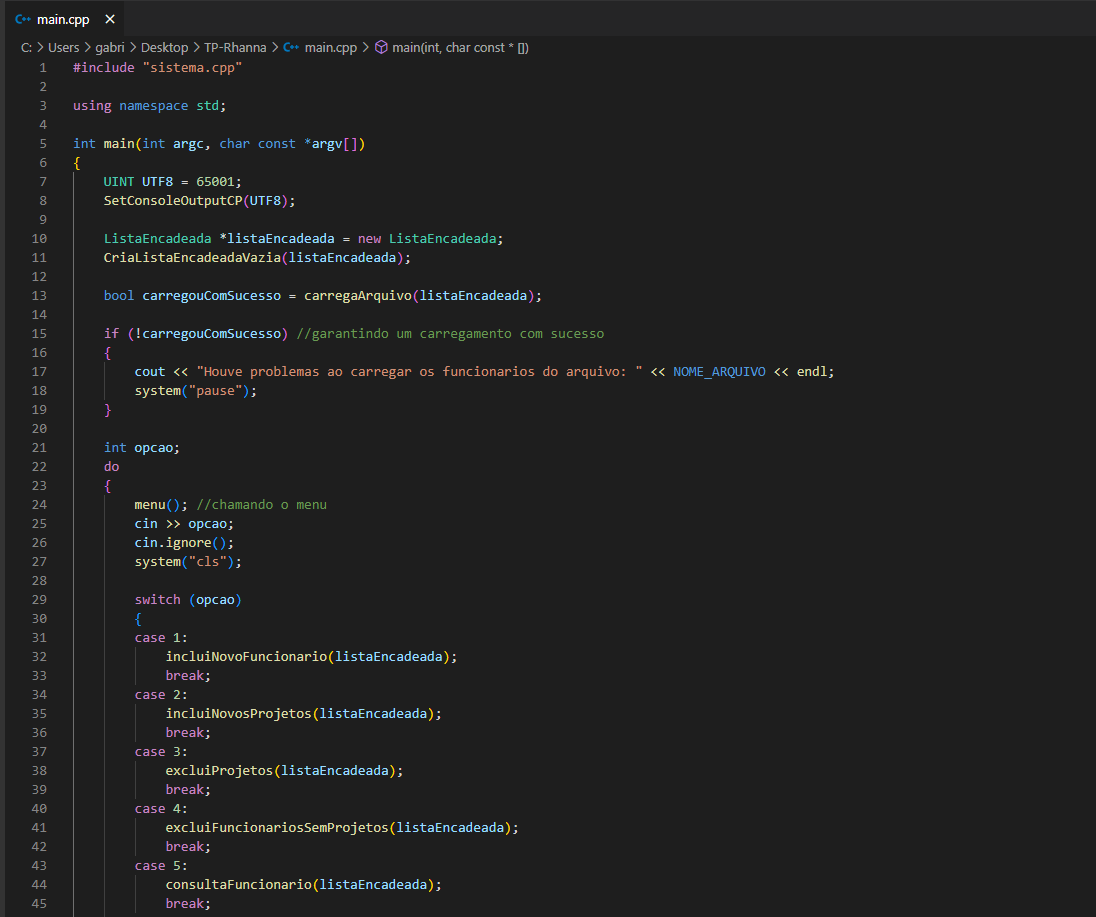
A próxima e última função, serve para calcular o valor do Contracheque dos funcionários com base no número de horas trabalhadas nesses projetos (Apêndice G), para calcular o salário bruto (Apêndice H) e os descontos de inss e o da receita federal, mostrados na Figura 14.

Figura 14



Por fim temos a apresentação da Main.cpp na Figura 15, onde são chamadas todas as funções principais contidas no Menu, e onde é feito o carregamento e salvamento do arquivo.

Figura 15



# CONCLUSÃO

Ao longo do trabalho encontrei dificuldades para interpretar o que estava sendo pedido, fui a monitoria para sanar minhas dúvidas, conversei com poucos colegas de classe, para me auxiliarem na interpretação e na escrita do código do trabalho, fiz uso dos slides disponibilizados pelo professor para sanar dúvidas a respeito das TADs (Tipos Abstratos de Dados), criação de funções de manipulação de arquivos, e também utilizei as informações da biblioteca, sites e fóruns.

Coloquei o trabalho em modularização separando-o em 5 arquivos, 2 arquivos \*.cpp, o sistema.cpp, com as funções e a main.cpp com a execução das funções, 1 arquivo \*.hpp, o sistema.hpp, onde tinha as structs e o cabeçalho das funções, 1 arquivo \*.bin, listaDeFuncionarios.bin onde os funcionários eram salvos, e 1 arquivo \*.exe, a main.exe, que seria o executável do código, onde as funções compilavam.

Creio eu que meu desenvolvimento apesar das dificuldades de interpretação, foram ótimos, aprendi coisas novas como funções que retornam um item, manipular arquivos, usar duas Estruturas de Dados para compor 1 só. Achei bastante semelhante ao que fiz no Ensino Médio Técnico Integrado, onde fizemos um sistema para atender à uma pequena empresa. Acho que consegui atingir meus objetivos, e explicar bem o que cada função faz, apresentar conhecimentos em lista encadeada, lista sequencial, manipulação de arquivos, e aplicar isso em um software, que atendesse o mínimo que uma empresa precisaria. Estou feliz com o resultado, e ele atendeu às minhas expectativas além do que eu esperava.



# REFERÊNCIAS

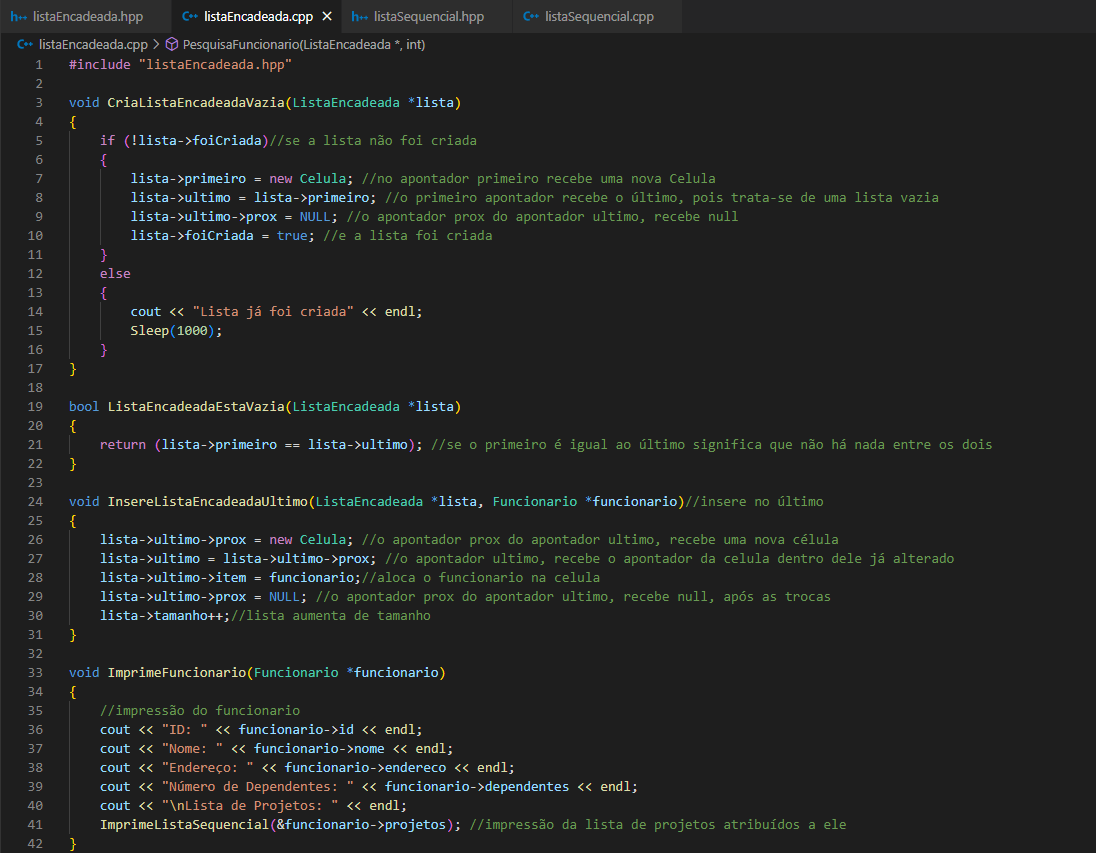
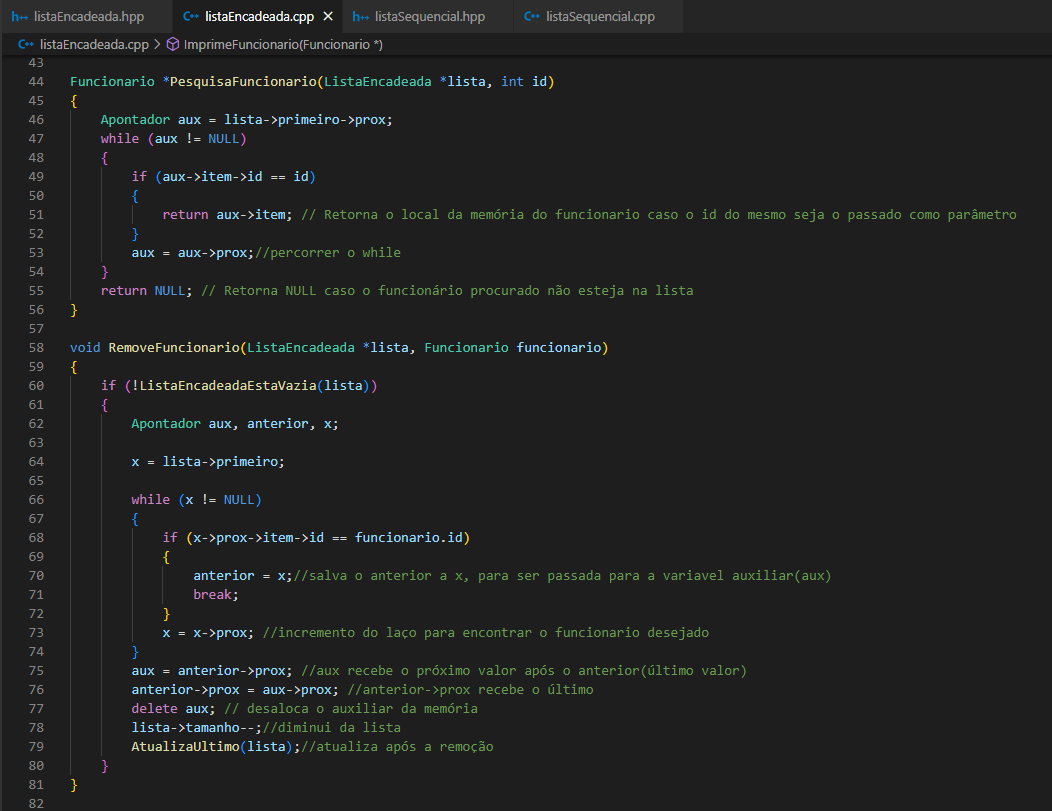
TRINDADE. Eduardo. Algoritmos e Estrutura de Dados – Arquivos. 2022. Apresentação PDF. Disponível em: [https://ead.ifmg.edu.br/saojoaoevangelista/pluginfile.php/146487/mod\_resource/content/1/Aula%203%20-%20Arquivos.pdf](https://ead.ifmg.edu.br/saojoaoevangelista/pluginfile.php/146487/mod_resource/content/1/Aula%203%20-%20Arquivos.pdfv) . Acesso em: 25 de setembro de 2022.

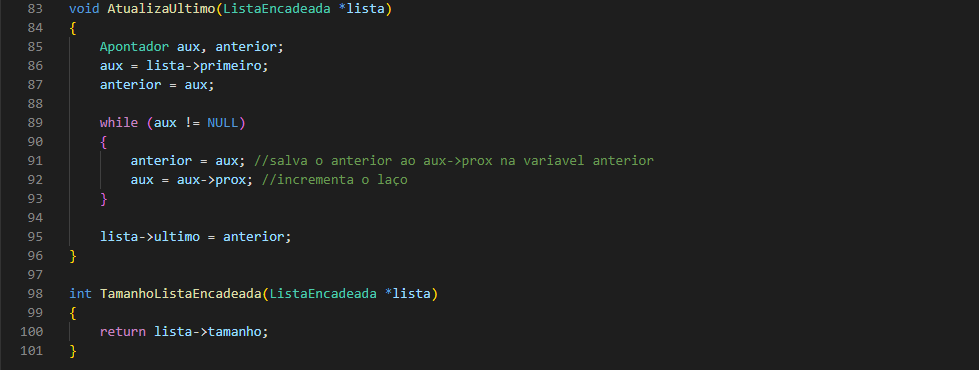
TRINDADE. Eduardo. Algoritmos e Estrutura de Dados - Listas. 2022. Apresentação PDF. Disponível em: <https://ead.ifmg.edu.br/saojoaoevangelista/pluginfile.php/146499/mod_resource/content/1/Aula%206%20-%20Listas.pdf> . Acesso em: 25 de setembro de 2022.

TRINDADE. Eduardo. Algoritmos e Estrutura de Dados - Listas utilizando Ponteiro. 2022. Apresentação PDF. Disponível em: <https://ead.ifmg.edu.br/saojoaoevangelista/pluginfile.php/146500/mod_resource/content/1/Aula%207%20-%20Listas%20Encadeadas.pdf> . Acesso em: 25 de setembro de 2022.

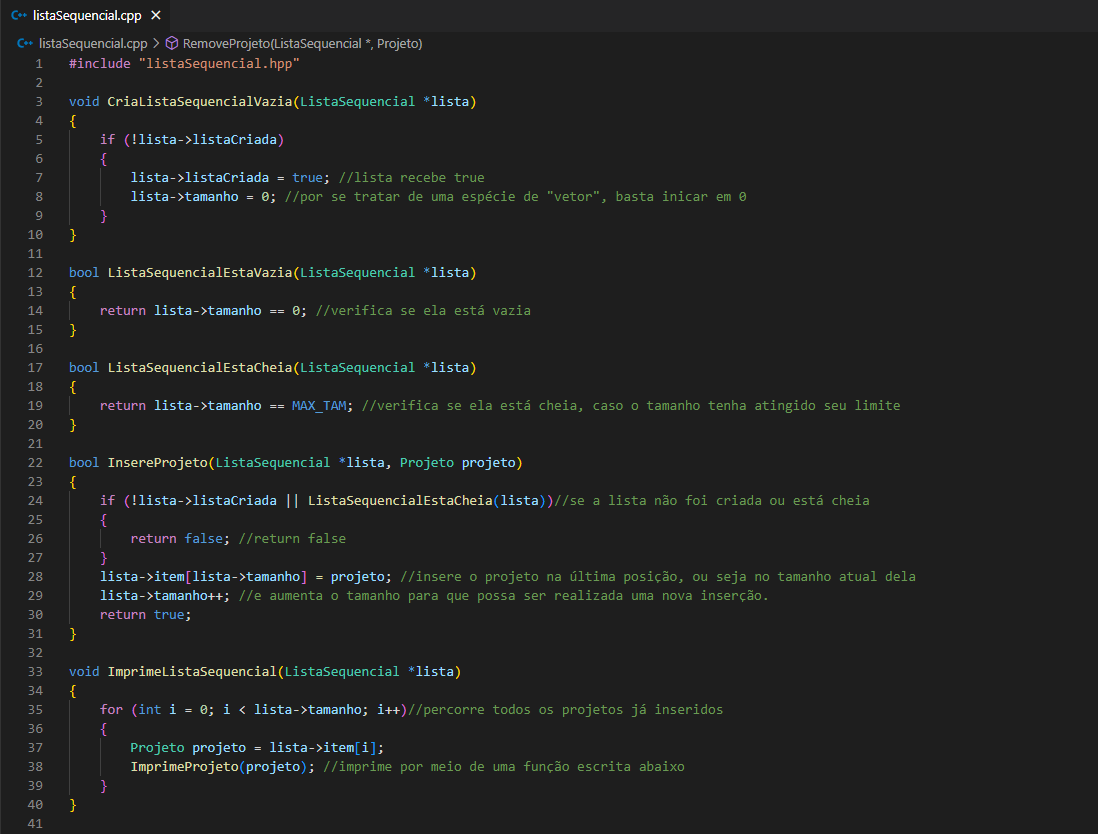
Programar em C++/Entrada e saída de dados 2. **WIKI LIVROS**, 16/04/2020.Disponível em: <https://pt.wikibooks.org/wiki/Programar_em_C%2B%2B/Entrada_e_saída_de_dados_2#:~:text=biblioteca%20padrão%20fstreamEditar&text=Esta%20biblioteca%20define%203%20novos,"in%20from%20a%20file"%20>. Acesso em: 25 de setembro de 2022.

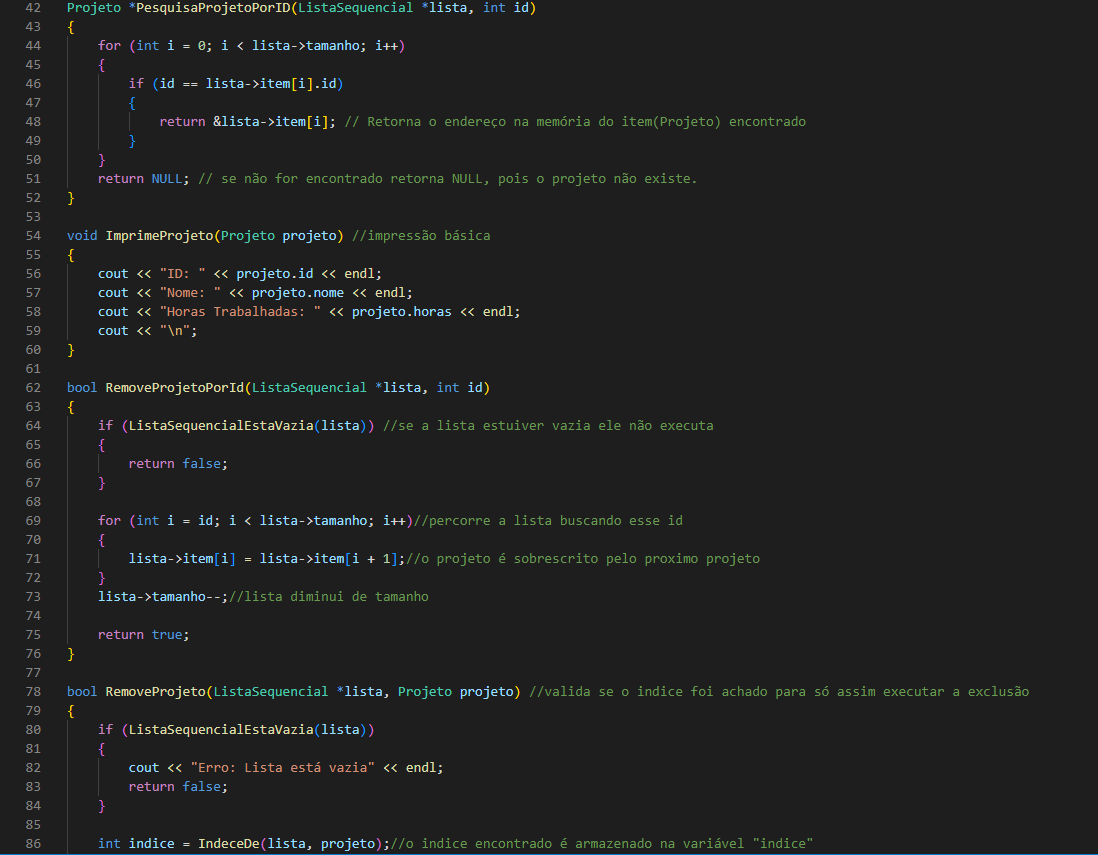
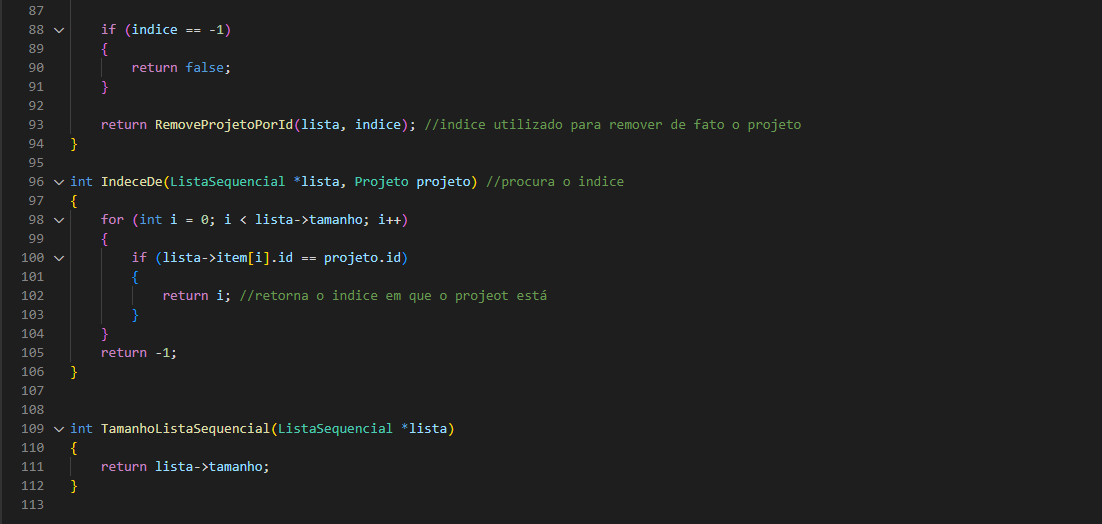
# APÊNDICES

1. 
   1. APÊNDICE A Funções para manipulação de lista Encadeada

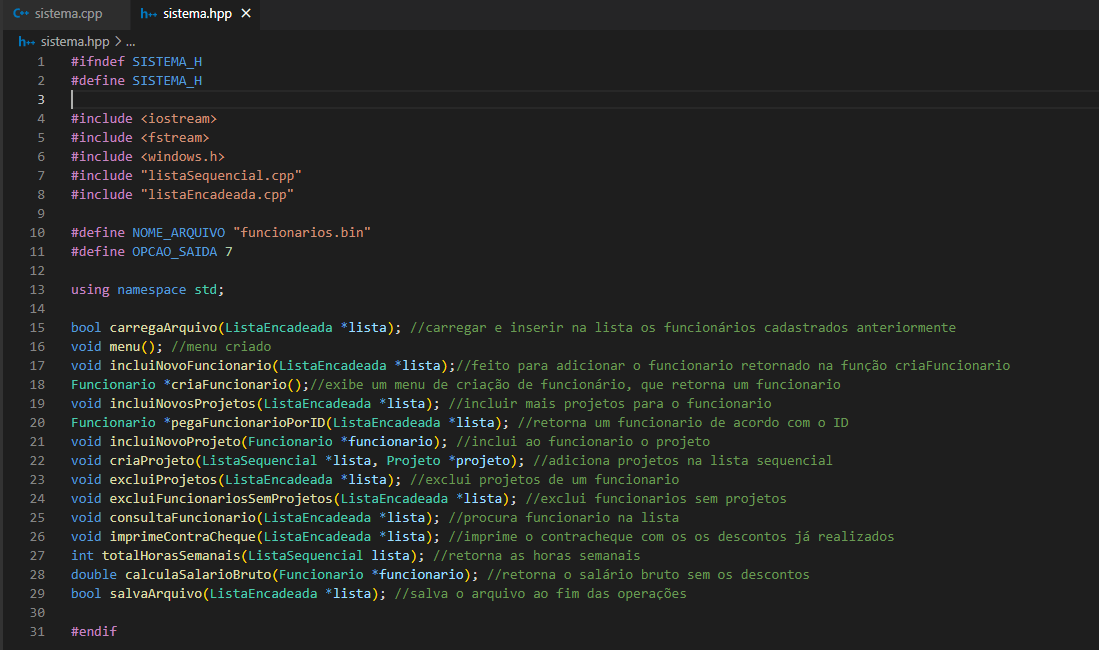


* 1. APÊNDICE B – Funções para a manipulação de Lista Sequencial

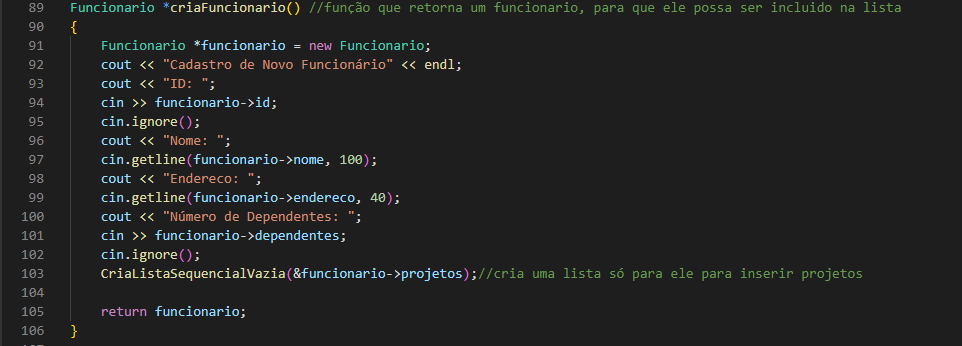


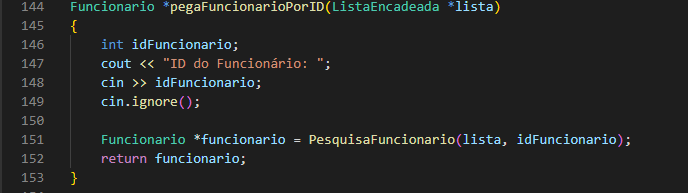
* 1. APÊNDICE C – Sistema.hpp



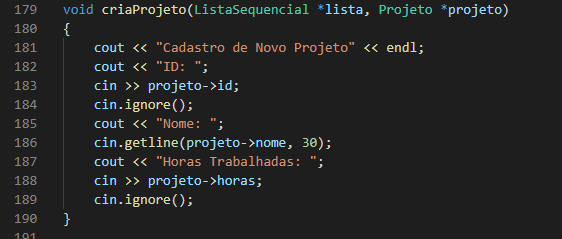
* 1. APÊNDICE D – Criação de funcionário



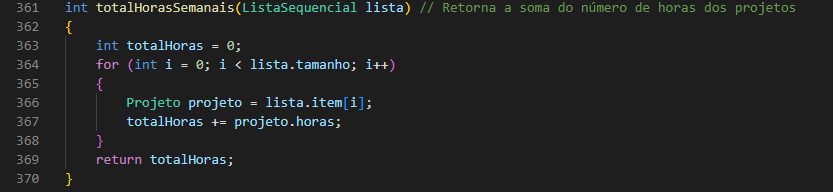
* 1. APÊNDICE E – Pesquisa ID repetido



* 1. APÊNDICE F – Função de criação de projeto



* 1. APÊNDICE G – Cálculo de Horas semanais trabalhadas.



* 1. APÊNDICE H – Cálculo de Salário Bruto

